

CLAIMS

1. One or more computer-readable media comprising a plurality of data structures, the plurality of data structures comprising:

a code unit having executable instructions;

a rewriter list identifying at least one rewriter; and

one or more rewriters that include the at least one rewriter, each rewriter capable of rewriting the code unit.

2. One or more computer-readable media as recited in claim 1, further comprising a rewrite manager, the rewrite manager having executable instructions configured to access the at least one rewriter from the one or more rewriters and to execute the at least one rewriter against the code unit, generating a rewritten code unit.

3. One or more computer-readable media as recited in claim 1, further comprising a cache, the rewrite manager further configured to store the rewritten code unit in the cache.

4. One or more computer-readable media as recited in claim 1, wherein the rewriter list is selected from the group comprising:

a list of one or more custom attributes within the code unit;

a list within a security policy;

a list within an installation tool;

a list within a configuration file; and

a list within an XML (Extensible Markup Language) file.

5. One or more computer-readable media as recited in claim 2, wherein the rewrite manager is a module selected from the group comprising:

- a stand-alone module;
- an operating system module;
- an execution environment module;
- a JIT compiler module;
- a source code compiler module; and
- an installation tool configured to install the code unit.

6. A computer comprising the one or more computer-readable media recited in claim 1, the computer selected from the group comprising:

- a developer computer configured to create the code unit;
- an intermediate computer configured to deploy the code unit; and
- a deployment computer configured to execute the code unit.

7. One or more computer-readable media comprising a code unit, the code unit including executable instructions configured for:

- initiating a transformation of the code unit; and
- identifying one or more rewriters to implement the transformation.

8. One or more computer-readable media as recited in claim 7, wherein the one or more rewriters is a plurality of rewriters, each rewriter configured to implement a unique transformation of the code unit, the code unit including further

executable instructions configured for sequencing the plurality of rewriters to implement the unique transformations in a particular order.

9. One or more computer-readable media as recited in claim 7, the code unit including further executable instructions configured for guiding the one or more rewriters to implement the transformation against identified elements within the code unit.

10. One or more computer-readable media as recited in claim 7, wherein the initiating comprises:

identifying an environment in which the transformation should be implemented; and

initiating the transformation only if the code unit experiences the identified environment.

11. One or more computer-readable media as recited in claim 10, wherein the identifying an environment comprises:

identifying a source code compilation environment;

identifying a pre-execution-compilation environment;

identifying a compilation-on-installation environment;

identifying an execution environment; and

identifying an installation environment.

12. One or more computer-readable media as recited in claim 7, wherein the identifying one or more rewriters comprises listing the one or more rewriters within the code unit.

13. One or more computer-readable media as recited in claim 7, wherein the identifying one or more rewriters comprises identifying a file separate from the code unit that includes a list of the one or more rewriters.

14. One or more computer-readable media comprising computer-executable instructions configured for:

receiving a code unit having executable instructions;

determining at least one rewriter with which the code unit may be rewritten;

calling the at least one rewriter; and

executing the at least one rewriter against the code unit to generate a rewritten code unit.

15. One or more computer-readable media as recited in claim 14, comprising further computer-executable instructions configured for verifying trustworthiness of the code unit and the at least one rewriter prior to the executing.

16. One or more computer-readable media as recited in claim 15, wherein the verifying comprises authenticating a digital signature.

17. One or more computer-readable media as recited in claim 14, comprising further computer-executable instructions configured for storing the rewritten code unit in a cache.

18. One or more computer-readable media as recited in claim 17, comprising further computer-executable instructions configured for:

- receiving an instruction to execute the code unit;
- recognizing that the code unit has been rewritten;
- loading the rewritten code unit from the cache; and
- executing the rewritten code unit.

19. One or more computer-readable media as recited in claim 17, comprising further computer-executable instructions configured for:

- generating a digital signature for the rewritten code unit; and
- associating the digital signature with the rewritten code unit.

20. One or more computer-readable media as recited in claim 14, wherein the at least one rewriter is a plurality of rewriters, the one or more computer-readable media comprising further computer-executable instructions configured for sequencing the plurality of rewriters to rewrite the code unit in a particular rewrite order.

21. One or more computer-readable media as recited in claim 20, wherein the sequencing comprises:

- accessing a rewriter list; and

setting the rewrite order according to the rewriter list.

22. One or more computer-readable media as recited in claim 21, wherein the accessing comprises accessing the rewriter list in a location selected from the group comprising:

the code unit;

a separate file associated with the code unit; and

a system policy.

23. One or more computer-readable media as recited in claim 14, wherein the determining the at least one rewriter comprises reading a rewriter list.

24. One or more computer-readable media as recited in claim 14 embodied in a tool selected from the group comprising:

a source code compiler;

an installation tool;

a managed execution environment;

a stand-alone rewrite management tool; and

a JIT (just in time) compiler.

25. A computer comprising the one or more computer-readable media recited in claim 14, the computer selected from the group comprising:

a developer computer configured to develop the code unit;

an intermediate computer configured to install the code unit; and

a deployment computer configured to execute the code unit.

26. A computer comprising:
a code unit;
a composable set of rewriters, each rewriter configured to rewrite the code unit in a unique manner;
a rewrite manager configured to identify one or more rewriters from the composable set of rewriters and to execute the identified one or more rewriters against the code unit; and
a rewritten code unit generated by executing the identified one or more rewriters against the code unit.

27. A computer as recited in claim 26, further comprising a rewrite cache, the rewrite manager further configured to store the rewritten code unit in the rewrite cache.

28. A computer as recited in claim 26, further comprising a rewriter list from which the rewrite manager identifies the one or more rewriters to execute against the code unit.

29. A computer as recited in claim 28, wherein the rewriter list is a component selected from the group comprising:
a list of rewriters in the code unit;
a list of rewriters in a stand-alone file;
a list of rewriters in a security policy; and
a list of rewriters in an installation tool.

30. A computer as recited in claim 26, further comprising:
a first digital signature associated with the code unit; and
a set of second digital signatures, each second digital signature associated with a specific rewriter from the composable set of rewriters;

wherein the rewrite manager is further configured to determine if the code unit is trusted based on the first digital signature, determine if each rewriter from the identified one or more rewriters is trusted based on a corresponding second digital signature from the set of second digital signatures, and execute the identified one or more rewriters against the code unit only if both the code unit and each rewriter from the identified one or more rewriters are trusted.

31. A computer as recited in claim 30, further comprising a third digital signature associated by the rewrite manager with the rewritten code unit and configured to verify that the rewritten code unit is trusted.

39. A computer as recited in claim 26, wherein the rewrite manager is a component selected from the group comprising:

- a stand-alone rewrite module;
- a rewrite module configured as part of an operating system;
- a rewrite module configured as part of an installation tool; and
- a rewrite module configured as part of a security policy.

40. A computer as recited in claim 26, selected from the group comprising:

a development computer configured to develop the code unit;
an intermediate computer configured to install the code unit; and
a deployment computer configured to execute the code unit.

41. A method comprising:
receiving an executable code unit;
determining that the code unit needs to be rewritten;
determining one or more rewriters to rewrite the code unit; and
running the one or more rewriters against the code unit to generate a rewritten code unit.

42. A method as recited in claim 41, further comprising verifying trustworthiness of the code unit and the one or more rewriters prior to the running.

43. A method as recited in claim 42, wherein the verifying comprises authenticating a digital signature.

44. A method as recited in claim 41, further comprising storing the rewritten code unit in a cache.

45. A method as recited in claim 44, further comprising:
receiving an instruction to execute the code unit;
recognizing that the rewritten code unit is stored in the cache;
loading the rewritten code unit from the cache; and
executing the rewritten code unit.

46. A method as recited in claim 44, further comprising:
generating a digital signature for the rewritten code unit; and
associating the digital signature with the rewritten code unit.
47. A method as recited in claim 41, further comprising sequencing the
one or more rewriters to rewrite the code unit in a particular rewrite order.
48. A method as recited in claim 47, wherein the sequencing comprises:
accessing a rewriter list; and
setting the rewrite order according to the rewriter list.
49. A method as recited in claim 41, wherein the determining one or
more rewriters comprises reading a rewriter list.
50. A method as recited in claim 49, wherein the reading comprises
reading the rewriter list in a location selected from the group comprising:
the executable code unit; and
a separate file associated with the executable code unit.
51. A method comprising:
receiving a code unit;
determining that the code unit is to be rewritten by a rewriter;
determining if the code unit and the rewriter are trusted;

running the rewriter against the code unit to generate a rewritten code unit if the code unit and the rewriter are trusted;

storing the rewritten code unit in a cache.

52. A method as recited in claim 51, further comprising:
generating a digital signature for the rewritten code unit; and
attaching the digital signature to the rewritten code unit.

53. A method as recited in claim 52, further comprising:
receiving a call to execute the code unit;
recognizing that the code unit has been rewritten;
loading the rewritten code unit from the cache;
verifying the digital signature attached to the rewritten code unit;
and
executing the rewritten code unit if the verifying indicates that the rewritten code unit is secure.

54. A method as recited in claim 51, wherein the rewriter is an application compatibility rewriter and the determining that the code unit is to be rewritten comprises:

identifying the code unit as an application; and

consulting an application compatibility rewrite database to determine if any part of the application needs to be rewritten for compatibility with a current execution environment.